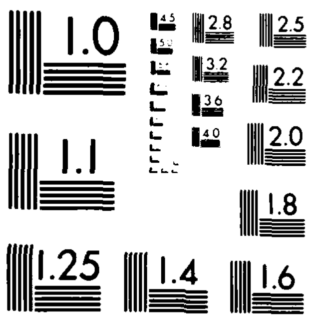
AD-A146 288 DESIGNING OPTIMAL AOQL SAMPLING PLANS A COMPUTERIZED APPROACH(U) FLORIDA UNIV GAINESVILLE DEPT OF INDUSTRIAL AND SYSTEMS ENGINEERING J E WALKER ET AL. JAN 84 1/1

UNCLASSIFIED RR-84-1 N00014-83-K-0093 F/G 12/1 NL

END
DATE
FILMED
10 84
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

AD-A146 288

Thi[illegible] [illegible] approved
for publi[illegible]
distribution [illegible] unlimited

84 09 11 063

DESIGNING OPTIMAL AOQL SAMPLING PLANS
A COMPUTERIZED APPROACH

Research Report No. 84-1

by

Jo Ellen Walker
and
Richard S. Leavenworth

January 1984

Department of Industrial and Systems Engineering
University of Florida
Gainesville, Florida 32611

This research was supported by the U.S. Department of the Navy, Office of Naval Research, Under Contract N0014-83-K-0093.

FINDINGS OF THIS REPORT ARE NOT TO BE CONSTRUED AS AN OFFICIAL DEPARTMENT OF THE NAVY POSITION UNLESS SO DESIGNATED BY OTHER AUTHORIZED DOCUMENTS.

TABLE OF CONTENTS

ABSTRACT

This study provides an algorithm and computer program, coded in Fortran IV, to develop double sampling acceptance plans for attributes which satisfy a specified Average Outgoing Quality Limit (AOQL) and minimize the Average Fraction ~~Instected (AFI)~~ at a specified quality level, P_1. An analysis of the steps in the algorithm is provided and comparison is made among the results of program runs and Dodge-Romig AOQL and MIL-STD-105D plans.

OBJECTIVE

This study develops a computer based algorithm to derive double sampling plans which minimize the Average Fraction Inspected, AFI(p1), at a designated quality level, p1. The AFI is defined as:

$$AFI = 1/N\ ((L(p)*ns) + N*(1-L(p)): \qquad \text{Single Sampling} \quad (1)$$

and,

$$AFI = 1/N\ (L(p)*n1) + Pa(n2)*n2 + N*(1-L(p)); \qquad \text{Double Sampling} \quad (2)$$

L(p) is the appropriate formulation for the probability of acceptance as a function of the incoming quality level, p. This study employs the binomial distribution which, for the single sampling case, yields:

$$L(p) = \sum_{d=0}^{c} \binom{n}{d} p^{d} (1-p)^{n-d} \qquad (3)$$

where n is the sample size and c the acceptance number.

For Double Sampling,

$$L(p) = \sum_{d_1=0}^{c_1} \binom{n_1}{d_1} p^{d_1} (1-p)^{n_1-d_1}$$

$$+ \sum_{d_1=c_1+1}^{r_1-1} \sum_{d_2=0}^{c_2-d_1} \binom{n_1}{d_1}\binom{n_2}{d_2} p^{d_1+d_2} (1-p)^{n_1+n_2-d_1-d_2} \qquad (4)$$

where n1 is the first sample size of a double sampling plan, n2 is the second sample size, c1, and c2 are the acceptance numbers on the first and second samples, respectively, and r1 is the rejection number of the first sample. The rejection number on the second sample, r2, always is c2+1.

Pa(n1) is defined as the probability of acceptance on n1 given as

$$Pa(n1) = \sum_{d_1=0}^{c_1} \binom{n_1}{d_1} p^{d_1}(1-p)^{n_1-d_1} \tag{5}$$

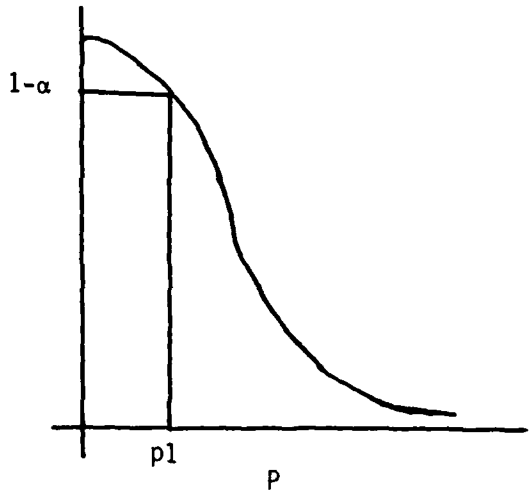
Pa(n2) is the probability of acceptance on n2 given as

$$Pa(n2) = \sum_{d_1=c_1+1}^{r_1-1} \quad \sum_{d_2=0}^{c_2-d_1} \binom{n_1}{d_1}\binom{n_2}{d_2} p^{d_1+d_2}(1-p)^{n_1+n_2-d_1-d_2} \tag{6}$$

Feasible sampling plans are those which strictly satisfy:

$$L(p1) > 1-\alpha \tag{7}$$

where p1 is the process average or the quality level considered acceptable for the purpose of acceptance sampling. The probability of acceptance must be least $1-\alpha$, where α is defined as the Producer's Risk. (p1 and α are specified design parameter).

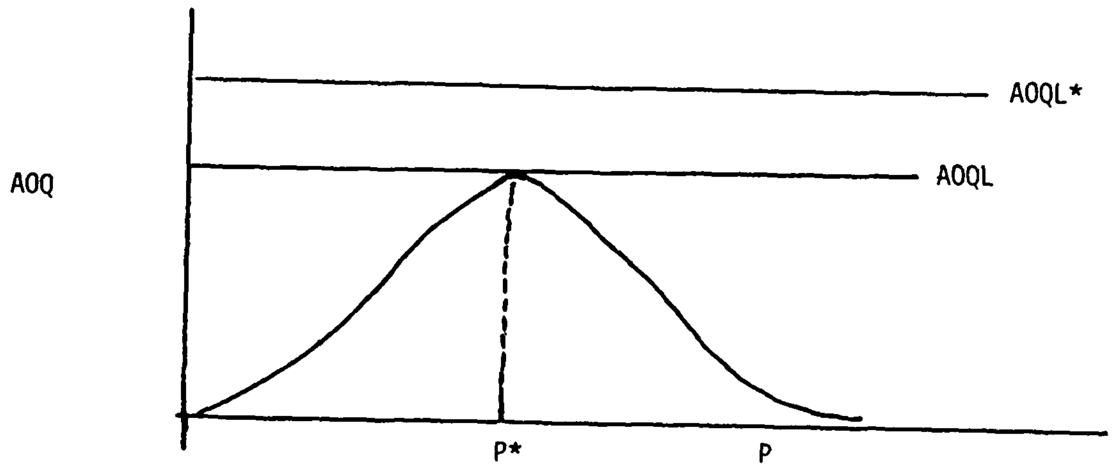


and AOQL < AOQL* (8)

where AOQL*: specified Average Outgoing Quality Limit

and AOQL: AOQ(p*) = p*(1-AFI(p*)) for a given sampling plan

p*: p value at which the AOQ function reaches its maximum value



It is useful at this point to differentiate between design parameters and acceptance sampling plan parameters. The design parameters for either a single sampling or double sampling plan are the required input values, p1

and α, the so-called Producer's Risk Point, and the required maximum AOQL, AOQL*. The plan parameters are the outputs of the programmed algorithm, ns and c for a single sampling plan and n1, n2, c1, c2, and r1, for a double sampling plan.

PROBLEM FORMULATION

Initially, a single sampling plan is found such that the AFI(p1) is minimized. The minimum (ns,c) combination satisfying AOQL $<$ AOQL* and satisfying L(p1) $>$ 1-α yields the optimal single sampling plan which is designated (n*,c*).

The bounds on the parameters of the double sampling plan are based on these results from deriving the single sampling plan. The first sample, n1, must satisfy

$$c2 < n1 < n^* \tag{9}$$

$$c2 > c^* \tag{10}$$

$$c1 < c2-1 \tag{11}$$

and $$B\,(n1,c2,p1) > 1-\alpha \tag{12}$$

where c1 and c2 are the acceptance numbers of the first and second samples, respectively, of a double sampling plan, and B(n,c,p) is the cumulative probability for a binomial distribution as given in equation (3). For purposes of this study, r1 has been made equal to r2, that is, c2+1, following the Dodge-Romig schemes. The second sample, n2, must satisfy:

$$n2 > n^*-n1 \tag{13}$$

and

$$BB(n1,n2,c1,c2,p1) > 1-\alpha \tag{14}$$

where BB(n1,n2,c1,c2,p) is the double cumulative probability of a binomial distribution as given in equation (4).

The objective is to minimize AFI(p1) subject to:

$L(p1) > 1-\alpha$

$AOQL < AOQL^*$

For each feasible c1,c2 combination, an optimal n1,n2 combination is found such that the AFI(p1) is minimized. The minimum AFI from all c1,c2 combinations yields the optimal sampling plan, n1*, n2*, c1*, c2*.

ALGOROTHM

The algorithm is divided into two parts, one for single sampling and one for double sampling. The bounds on some parameters of the double sampling plan are based on the results of deriving a corresponding single sampling plan, initially the optimal single sampling plan. Thus, to simplify discussion of the algorithm, the two parts are treated separately.

<u>Deriving the single sampling plan, (ns,c).</u>

1. Initialization: ns=1, c=0. Input α (> 0), p1, AOQL* ($>$ p1).
2. Compute B(ns,c,p1).
3. Test: Is B(ns,c,p1) $> 1-\alpha$?
    (a) If so, go to step 4.
    (b) If not so, increment c, set ns=c+1, and go to step 2.
    (If the first value of B(ns,c,p1) is not $> 1-\alpha$, then no larger value of ns can satisfy the constraint.)
4. Compute the AOQL for ns,c.
5. Test: Is the AOQL $<$ AOQL*?
    (a) If so, go to step 6.
    (b) If not so, increment ns and go to step 4.
    ($c < ns < N$)
6. Compute B(ns,c,p1).
7. Test: Is B(ns,c,p1) $> 1-\alpha$?
    (a) If so, go to step 8.
    (b) If not so, increment c, set ns=c+1, and go to step 4.
8. Compute the AFI(p1). Program outputs ns and c along with the AFI(p1) and actual AOQL after double sampling algorithm.

9. Set n*=ns,c*=c, go to double sampling algorithm. (Note - in future cycles step 8 is not employed.)

Deriving the Double Sampling Plan (n1,n2,c1,c2)

1. Initialization: Set c2=c*, c1=0, n1=c2+1, and n2=n*-n1; go to step 4. (B(n1=c2+1,c2,p1) ≥ 1-α is always satisfied.)
2. Compute B(n1,c2,p1).
3. Test: Is B(n1,c2,p1) ≥ 1-α ?
   (a) If so, go to step 4.
   (b) If not so, increment c2, set c=c2, set n=n*, go to step 2 of single sampling algorithm. (If the first value of B(n1,c2,p1) is not ≥ 1-α, then no larger value of n1 can satisfy the constraint.)
4. Compute BB(n1,n2,c1,c2,p1)
5. Test: Is BB(n1,n2,c1,c2,p1) ≥ 1-α ?
   (a) If so, go to step 6
   (b) If not so, increment c1 and go to step 4. (No larger value of n2 will satisfy the constraint.)
6. Compute the AOQL for the plan (n1,n2,c1,c2)
7. Test: AOQL(n1,n2,c1,c2) ≤ AOQL*
   (a) If so, go to step 8
   (b) If not so,increment n2 and go to step 6. (AOQL values decrease with increasing values of n2.)
8. Compute BB(n1,n2,c1,c2,p1)

9 Test: Is BB(n1,n2,c1,c2,p1) ≥ 1-α ?
   (a) If so, go to step 10.
   (b) If not so, go to step 9a.

9a. Test: Does n1=n*?

(a) If so, increment c1, set n1=c2+1, n2=n*-n1, and go to step 2.

(b) If not so, increment n1, set n2=n*-n1, and go to step 2.

10. Compute AFI(n1,n2,c1,c2,p1).

11. Test: is AFI(current plan) < AFI(previous plan)?

(a) If so, store AFI (current plan) and its plan parameters for the set value of c2 and go to step 12.

(b) If not so, retain AFI (previous plan) and its plan parameters for the set value of c2 and go to step 13.

12. Test: Does n1=n*?

(a) If so, increment c1, set n1=c2+1, n2=n*-n1, and go to step 2.

(b) If not so, increment n1, set n2=n*-n1, and go to step 2.

13. Test: Does c1=c2-1?

(a) If so, go to step 14.

(b) If not so, increment c1, set n1=c2+1, n2=n*-n1, and go to step 2.

14. Test: Is AFI(n1,n2,c1,c2,p1) < AFI(n1,n2,c1,c2-1,p1)?

(a) If so, retain AFI (current plan) and its plan parameters, increment c2, set c=c2, n=n*, and go to step 2 of the single sampling algorithm.

(b) If not so, previous plan AFI and its parameters (n1,n2,c1,c2) is optimal.

ANALYSIS OF THE ALGORITHM

The L(p1) decreases with increasing values of n. This result reduces the computational time greatly. As n1 increases, B(n1 2,p1) decreases (Figure 1). Once the L(p1) constraint i- violated, c2 is incremented rather than computing B(n1,c2,p1) for all n1 < n* for the given c2. Also, since the double cumulative probabilities decrease with increasing values of n2, the next n1 value is generated once BB(n1,n2,c1,c2,p1) < 1-α instead of continuing the computation for n2 < N-n1 (Figure 2).

Table 1 perhaps best simplifies further discussion. The tabular results are from a computer run wherein the specified parameters are p1=0.03, α=0.05, AOQL=0.05, and N=500. Each cell (block) contains the (n1,n2) pair and corresponding AFI(p1) values such that L(p1) > 1-α and AOQL < AOQL* for each feasible (c1,c2)combination, i.e. for c1 < c2. Table 2 shows the optimal plan within each cell (c1,c2 combination), the column minimum (fixed c2), and the optimal plan selected from the column minimums.

By studying the results as displayed in Table 1, the following conclusions are reached:

1. In each cell, the AFI(p1) increases with increasing n2 and fixed n1 (Figure 3). Therefore, it is necessary to find only the first n1,n2 combination satisfying the constraints rather than testing all feasible values of n2. The large list of the AFI values for all values of n2 and a given n1 is not included.

2. In each cell, the AFI is a convex function of each minimum n1,n2 combination (first feasible n1,n2 combination). The minimum sampling plan for each cell is one associated with the minimum of this function (Figure 4.) In Table 1, the minimum sampling plan of the first cell in the c2=2 column is n1=11, n2=16.

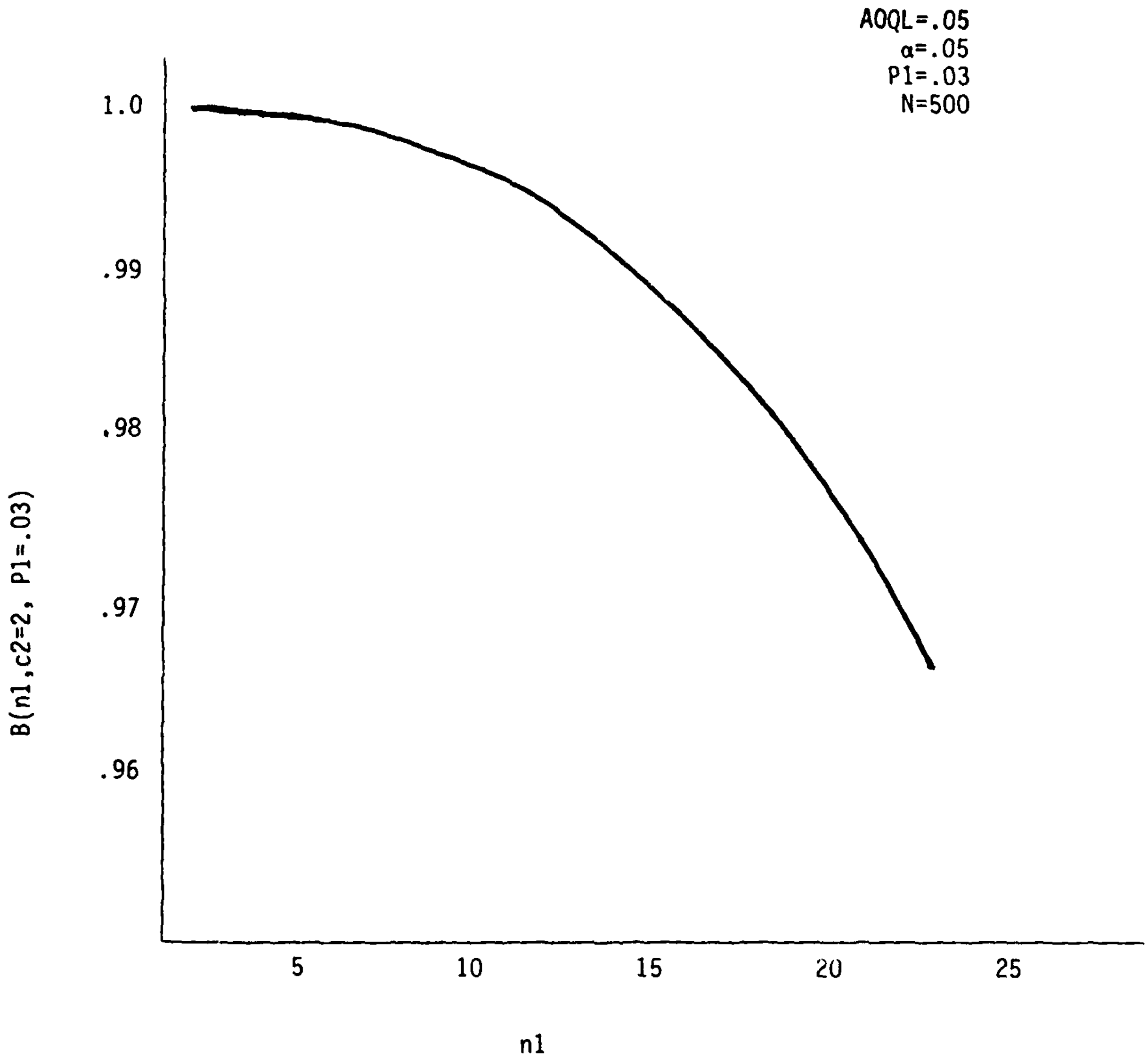


FIGURE 1. Binomial Probabilities as a Function of First Sample Size, n1.

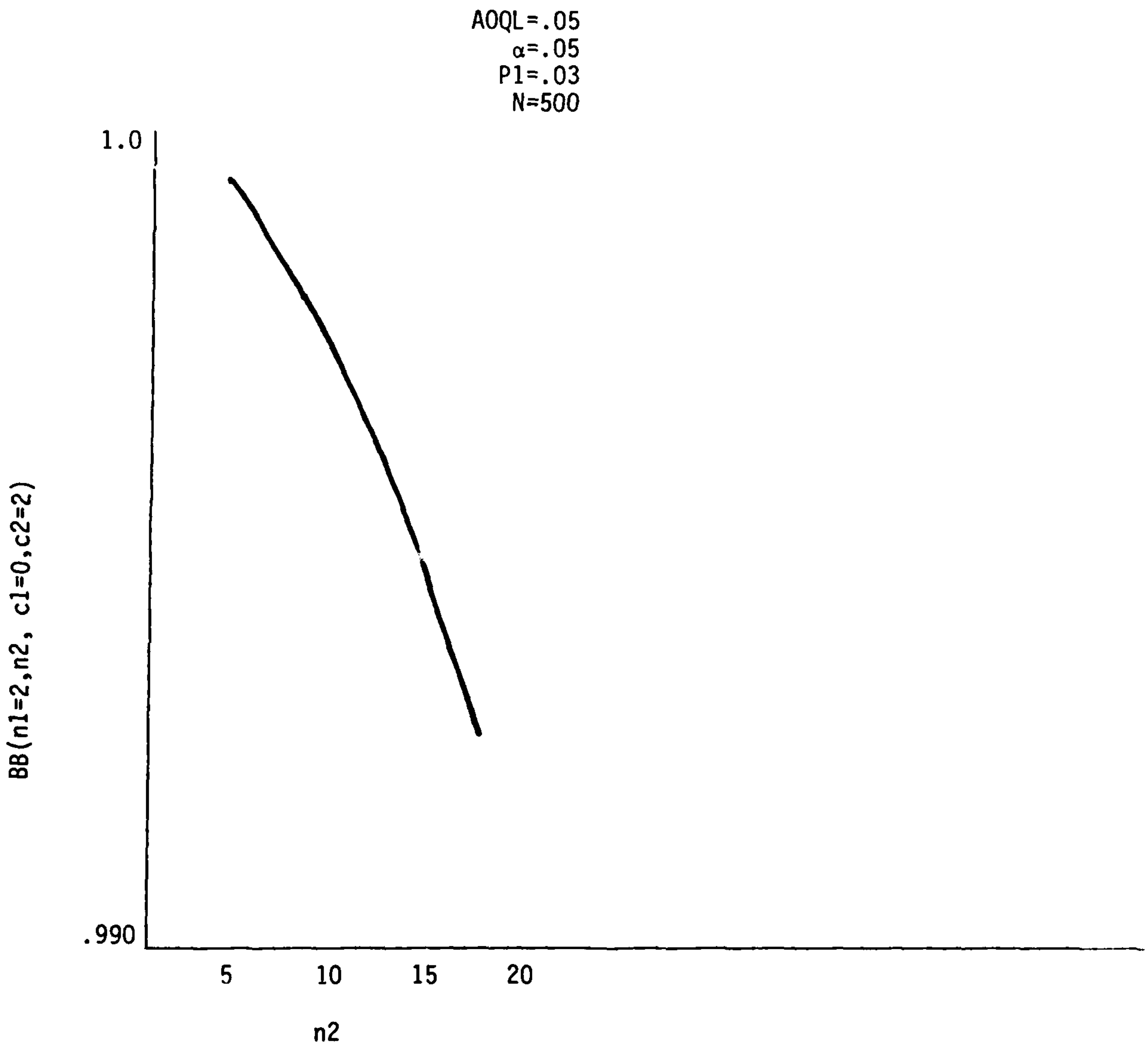


FIGURE 2. Double Binomial Probabilities as a Function of the Second Sample Size, n2.

TABLE 1

Schematic of Program Outputs, Double Sampling

	c2=2			c2=3			c2=4			c2=5		
	n1	n2	AFI(p1)	n1	n2	AFI(p1)	n1	n2	AFI(p1)	n1	n2	AFI(p1)
c1=0	9	23	.073063	7	54	.076090	7	67	.065837	7	80	.061384
	10	20	.072404	8	41	.062998	8	54	.057350	8	66	.055319
	11	16	.067187	9	35	.059519	9	47	.054957	9	58	.053936
	12	14	.067326	10	30	.056563	10	41	.052956	10	52	.053722
	13	13	.07038	11	26	.054611	11	37	.052887	11	47	.053912
	14	12	.073129	12	25	.057869	12	35	.054955	12	45	.056551
	⋮			⋮			⋮			⋮		
	20	16	.085011	31	6	.091863	41	6	.102421	51	6	.17795 cell
	n1	n2	AFI(p1)	n1	n2	AFI(p1)	n1	n2	AFI(p1)	n1	n2	AFI(p1)
c1=1	18	15	.082664	17	38	.071300	16	94	.083959	16	110	.077789
	19	12	.084097	18	30	.068552	17	53	.063678	17	67	.058314
	20	9	.084268	19	24	.066407	18	43	.060595	18	56	.057129
	21	7	.085656	20	21	.067226	19	37	.059908	19	49	.056860
	22	5	.086196	21	17	.066042	20	32	.059564	20	48	.057501
	23	4	.089133	22	15	.067319	21	28	.059740	21	39	.057907
	⋮			⋮			⋮			⋮		
	26	0	.091797	31	6	.087214	41	6	.098053	47	10	.108607
				n1	n2	AFI(p1)	n1	n2	AFI(p1)	n1	n2	AFI(p1)
c1=2				26	48	.084916	26	70	.080637	26	89	.077190
				27	23	.080726	27	41	.074589	27	57	.070797
				28	17	.081398	28	32	.073855	28	47	.070319
				29	13	.082497	29	26	.073923	29	40	.070588
				30	10	.083788	30	22	.074750	30	34	.070989
				⋮			⋮			⋮		
				33	4	.088216	41	6	.095250	51	6	.110683
							n1	n2	AFI(p1)	n1	n2	AFI(p1)
c1=3							37	27	.089033	37	46	.085970
							38	19	.090008	38	45	.086241
							39	14	.091277	39	29	.087349
							40	11	.093011	40	24	.088489
							41	8	.094413	41	20	.089759
							⋮			⋮		
							46	1	.103962	51	6	.109071
										n1	n2	AFI(p1)
c1=4										47	32	.102570
										48	21	.103790
										49	15	.105285
										50	12	.107183
										51	9	.108927

The minimums for each cell are given in Table 2.

TABLE 2

n1	n2	AFI(p1)	n1	n2	AFI(p1)	n1	n2	AFI(p1)	n1	n2	AFI(p1)
11	16	.067187	11	26	.054611	11	37	.052887	10	52	.053722
18	15	.082664	21	17	.066042	20	32	.059564	19	49	.056860
			27	23	.080726	28	32	.073855	28	47	.070319
						37	27	.089033	37	46	.085970
									47	32	.102570

Column Minimum	Column Minimum	Column Minimum	Column Minimum
n1=11 c1=0	n1=11 c1=0	n1=11 c1=0	n1=10 c1=0
n2=16 c2=2	n2=26 c2=3	n2=37 c2=4	n2=52 c2=5
AFI=.067187	AFI=.054611	AFI=.052887	AFI=.053722

Minimum double sampling plan: c1=0, c2=4, n1=11, n2=37, AFI(p1)=0.052887
AOQL=0.49464

Corresponding single sampling plan: c=2, n=26, AFI(p1)=0.091796, AOQL=0.049770

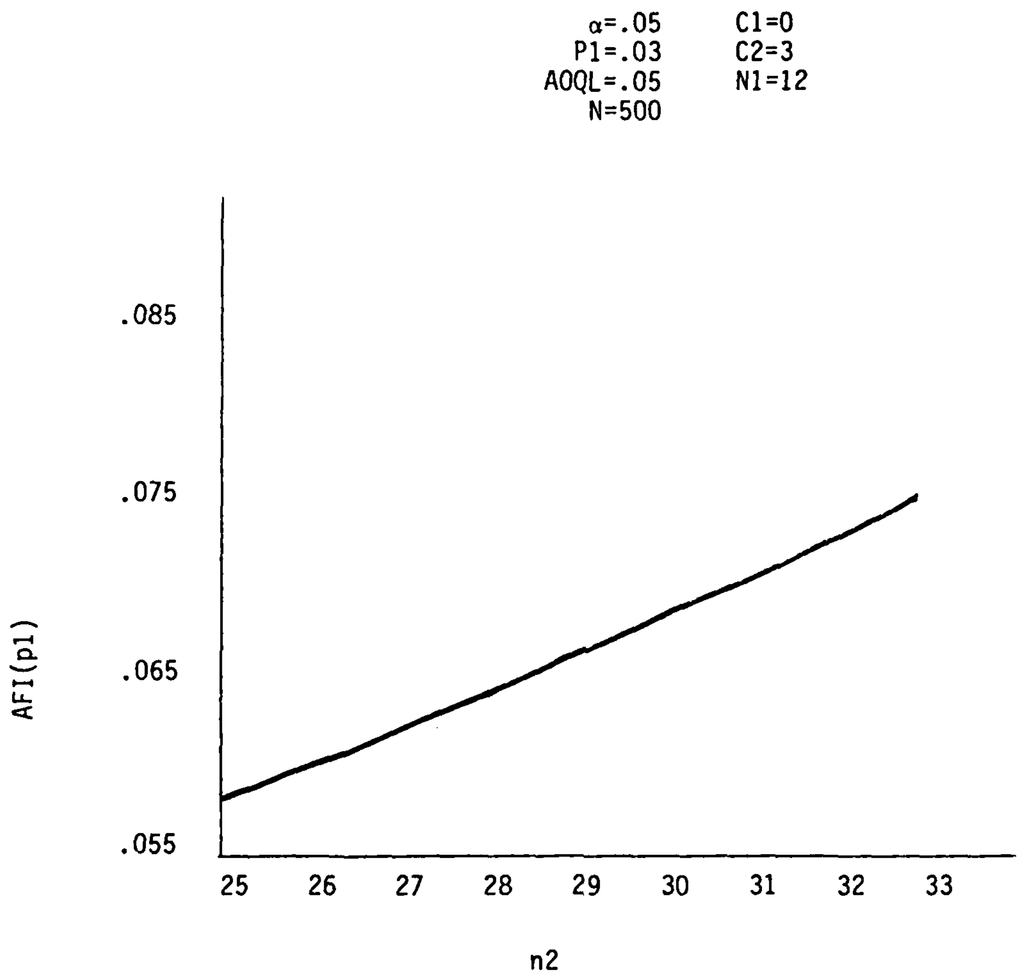


FIGURE 3. Change in AFI(p1) as a Function of n2.

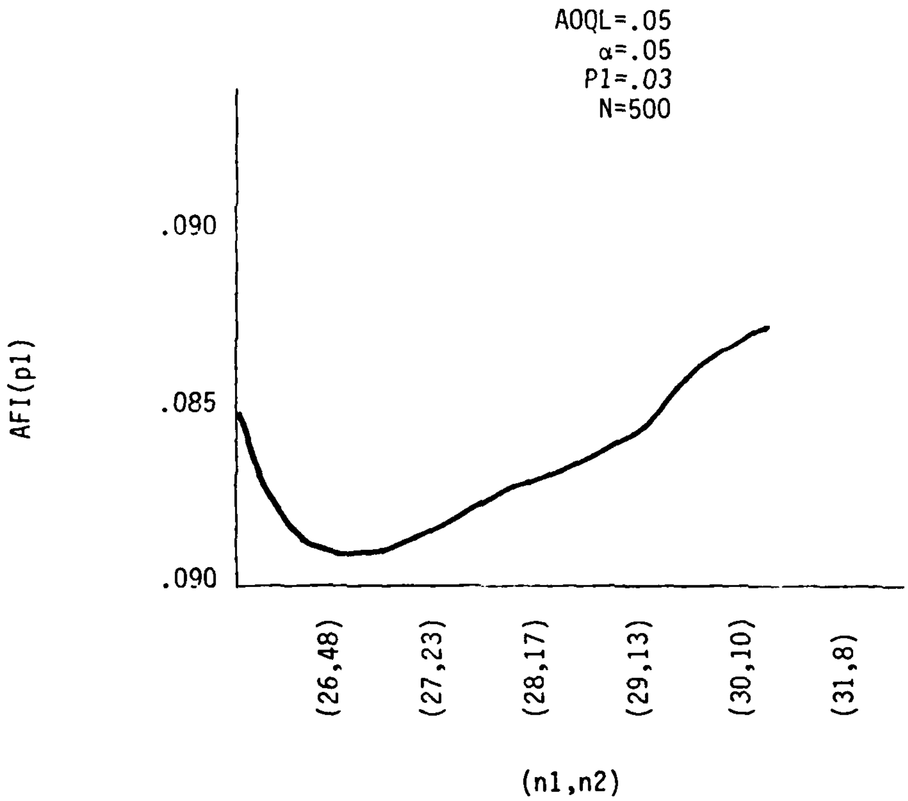


FIGURE 4. AFI(p1) as a Function of c1,c2 Cell Minimums

3. Each column gives cells for all feasible c1 values for a given c2. The cell AFI minimums are also convex in c1. Therefore, the minimum sampling plan for a given c2 is the one corresponding to the minimum of these cell minimums (Figure 5). For example, the minimum AFI of column 2 is 0.054611 and the corresponding sampling plan is c1=0, c2=3, n1=11, n2=26. (Note: It is not necessarily the case that the minimum always will occur at the lowest feasible value of c1. For example, for the parameters α=.023606, p1=.05, AOQL*=.05, and N=350, the minimum sampling plan is c1=1, c2=8, n1=21, n2=62. However, the minimum feasible value of c1 is 0.)

4. The minimums of the columns also follow this convex pattern. The column minimum AFI(p1) values in Table 1 corresponding to c2=2,3,4,5 are 0.072404, 0.0544611, 0.052887, 0.053722, respectively. This provides the minimum samling plan as well as a stopping criterion for the algorithm (Figure 6). Thus the minimum of the function, 0.052887, corresponds to the minimum sampling plan, c1=0, c2=4, n1=11, n2=37.

Single Sampling

The first feasible ns,c combination is found such that B(ns,c,p1) > 1-α. Since the binomial probabilites decrease as ns increases, the first infeasible value of ns indicates that all probabilities associated with larger values of ns for a given c are < 1-α. Thus, once a point becomes infeasible, c is incremented, and the process begins again.

For the first feasible ns value, the AOQL < AOQL* constraint is checked. This is accomplished in a subroutine which utilizes the Golden Section Method [3] to find the value of p,p*, which maximizes the AOQ

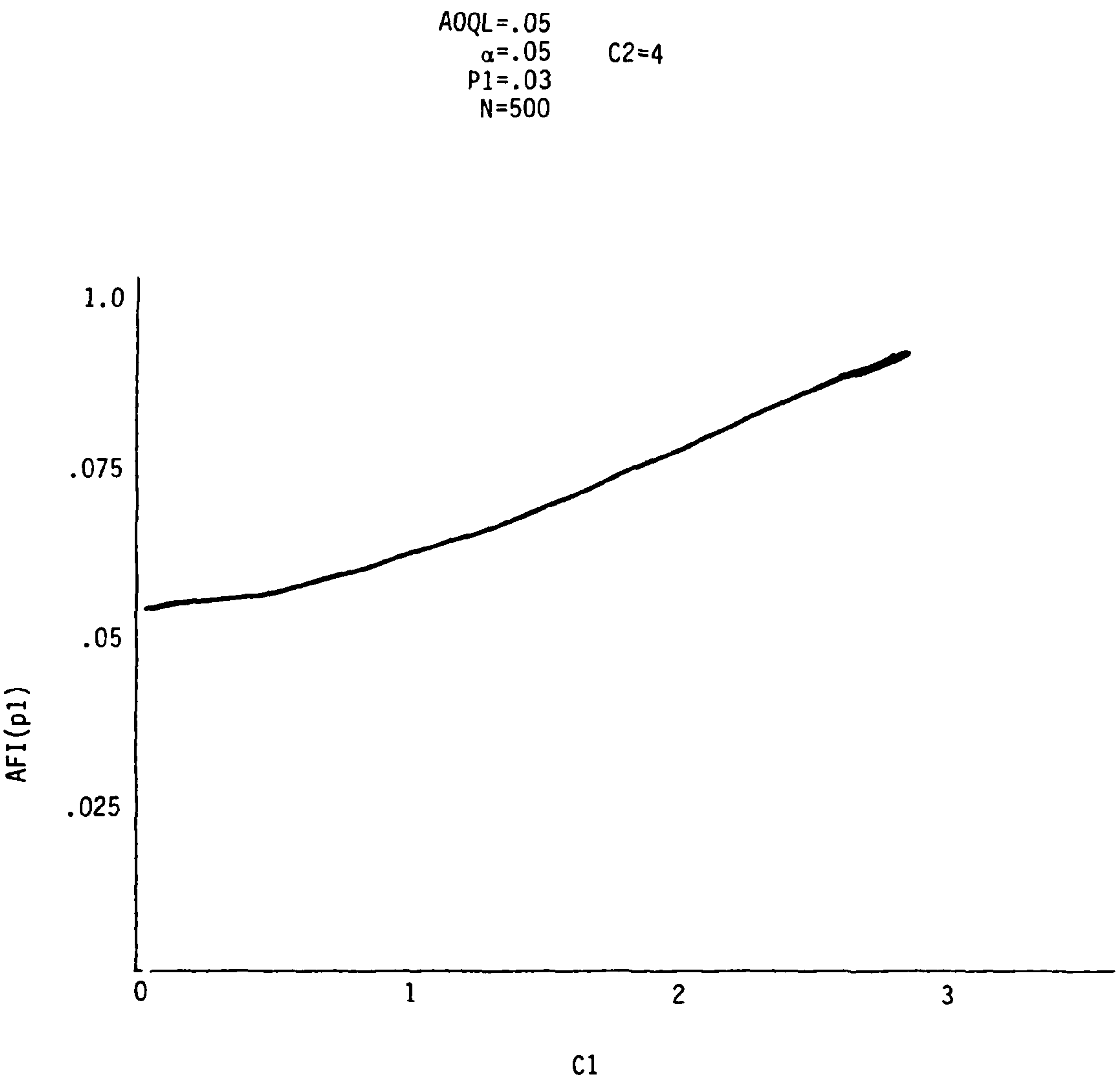


FIGURE 5. Minimum Cell AFI(p1) Values as a Function of c1.

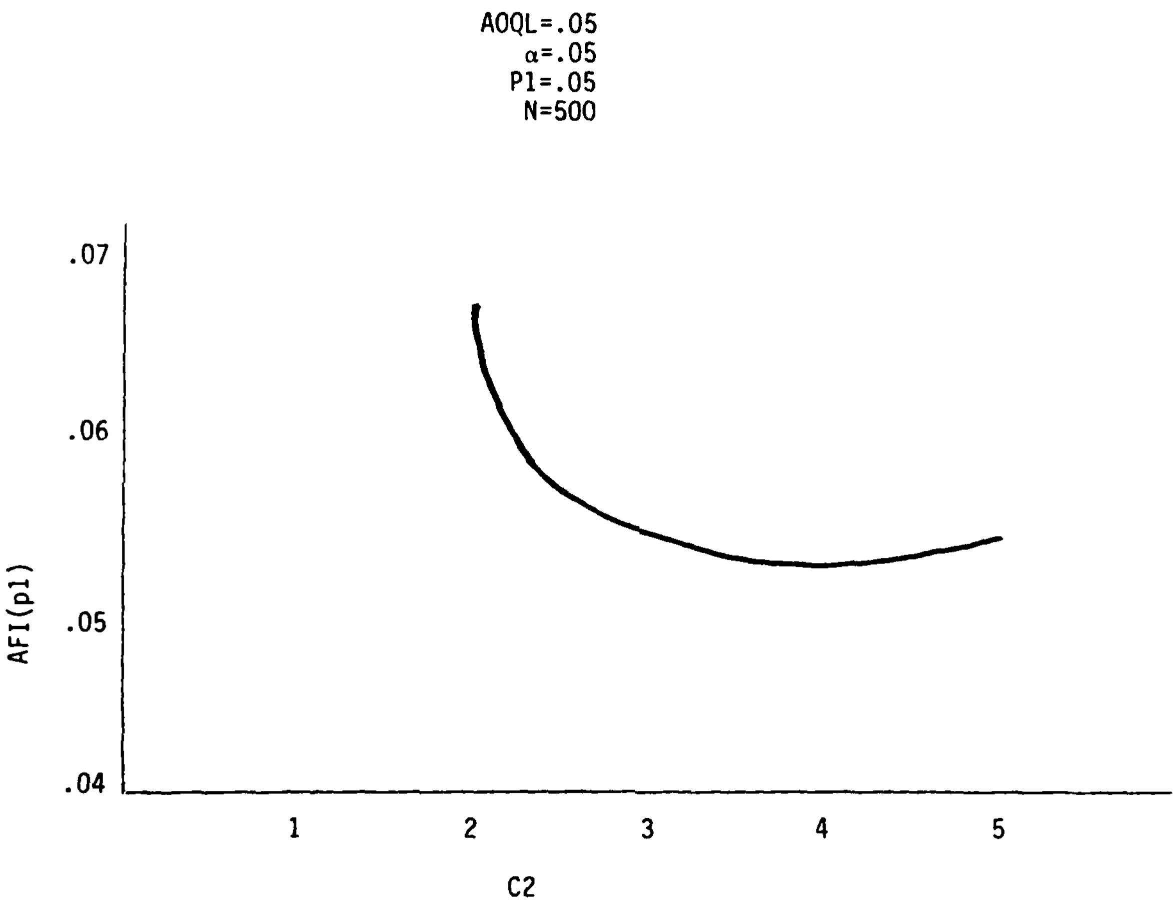


FIGURE 6. Value of AFI(p1) as a Function of c2.

function, i.e., the AOQL. p* may fall in the interval (0,1). The calculation of the required initial left bound (PLFT) and the initial right bound (PRT) are based on this allowable interval. In each iteration of the search, the AOQ values associated with PLFT and PRT are computed. If AOQ(PLFT) ≥ AOQ(PRT), PRT=PLFT. If not, PLFT=PRT, i.e., after comparing the AOQ values, only one of the p values is new. Thus only one additional computation of the AOQ is necessary. When the interval on p is less than 0.0001, the search ends, and for the necessary accuracy, it is assumed that p*=(PLFT+PRT)/2. If AOQL > AOQL*, the AFI is computed and the process begins again.

If AOQL < AOQL*, a bisection method is used to find the first ns value that satisfies the AOQ constraint. The initial bounds for the bisection are ns and N. If the AOQL value corresponding to N is greater than AOQL*, the bisection is omitted, c is incremented, and the process begins again. This is allowable since the AOQL values decrease with increasing values of ns (Figure 7). Use of the bisection method in this case appears to be very efficient. It requires log(2)N iterations rather than the N-ns possible iterations needed using a total enumeration method.

If the associated binomial probabilities are> 1-α, the AFI(p1) is computed. Otherwise, c is incremented, ns is set to the new c value, and the process begins again.

Double Sampling

The double sampling procedure is very similar. The size of the first sample, n1, ranges from c2+1 to n*. If B(n1,c2,p1) < 1-α, c2 is incremented. Since these cumulative binomial probabilites decrease with increasing n1, further enumeration for a given c2 is unnecessary. Whenever c2 is incremented, new bounds on n* must be computed using the single sampling algorithm. Since c2 corresponds to c*, a new n* satisfying the single

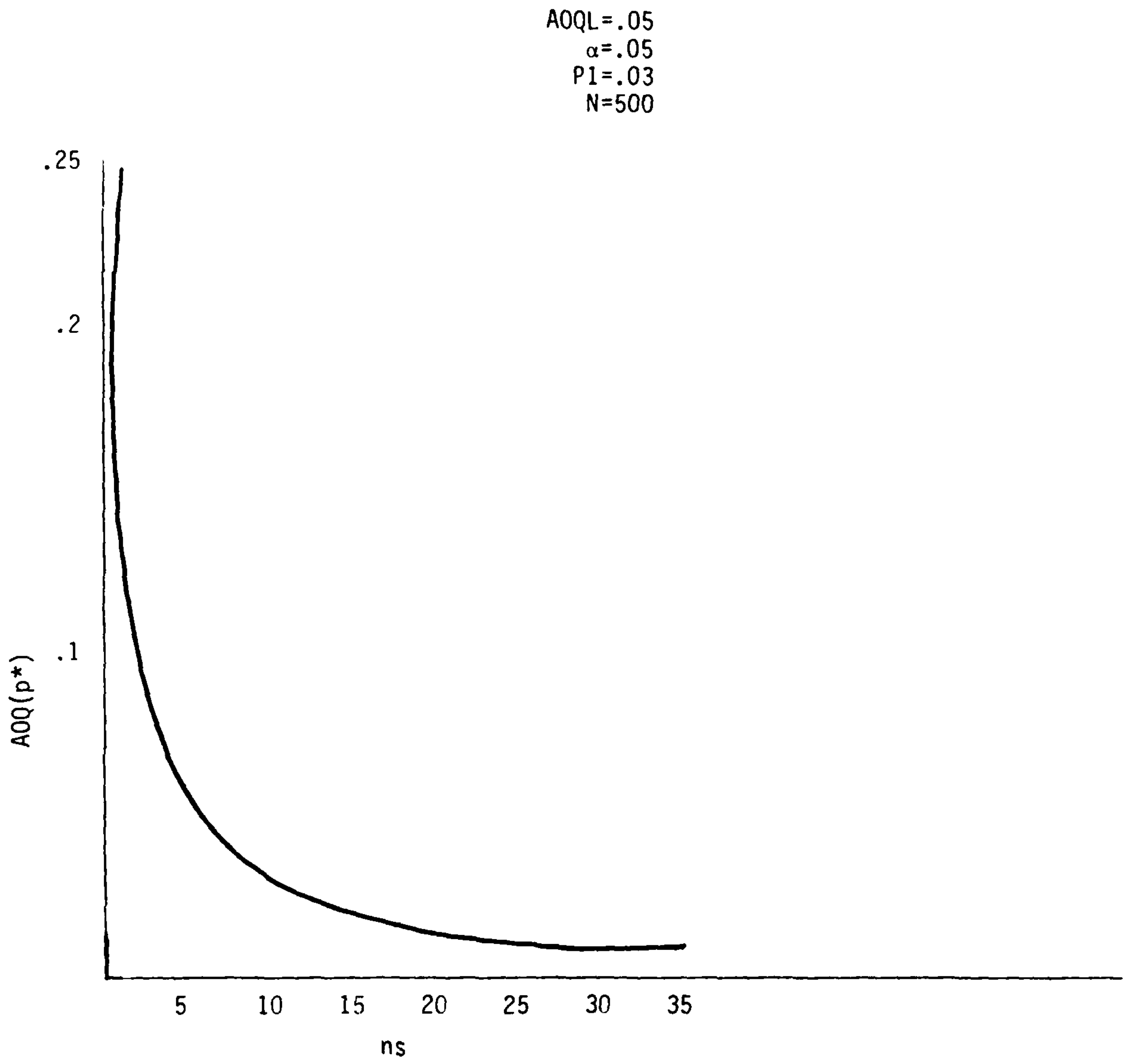


FIGURE 7. Values of AOQ as a Function of ns.

sampling criteria must be found. The double sampling algorithm continues with these new values of c2=c*, c2 < n1 < n*, and n2=n*-n1.

If B(n1,c2,p1) > 1-α, the corresponding feasible n2 > ns*-n1 is found. If BB(n1,n2,c1,c2,p1) < 1- α, c1 is incremented since the double cummulative probabilities decrease for the first feasible n1,n2 combinations of a given c1 (Figure 8). Otherwise, a bisection technique similar to that used for the single sampling case is used to find the lowest n2 value satisfying AOQL < AOQL*. If the cumulative double probability is greater than or equal to 1-α, the AFI(p1) is computed. If not, n1 is incremented and the process begins again. For a fixed n1, the first feasible (n1,n2) pair is the minimum of the AFI function.

The minimum sampling plan is one such that the AFI(p1) is minimized. First, the minimum for each c1 value, the cell minimum, is determined. This consists of the n1,n2 combination which yields the smallest AFI. As previously stated, the function is convex, however there is some variation since n2 does not decrease in uniform increments as n1 increases. In Table 1, when c1=1, c2=3, the AFI values decrease to 0.066407, increase to 0.067226, decrease to 0.066042, and finally increase to 0.067319 as n1 was increased. If the function were strictly convex, the computation for each cell would end upon the first increase in the value of AFI(p1). To account for this discontinuity, the algorithm continues for five additional increments of n1 to reasonably insure that a global minimum AFI(p1) is reached.

Next, the minimum for all feasible c1 values, the column minimum, is found. Due to the convexity of the AFI values corresponding to the c1 values, a column minimum is found after the first cell minimum increases.

The last step is to find the minimum AFI of all columns. The algorithm terminates after the minimum AFI corresponding to a c2 value, (column

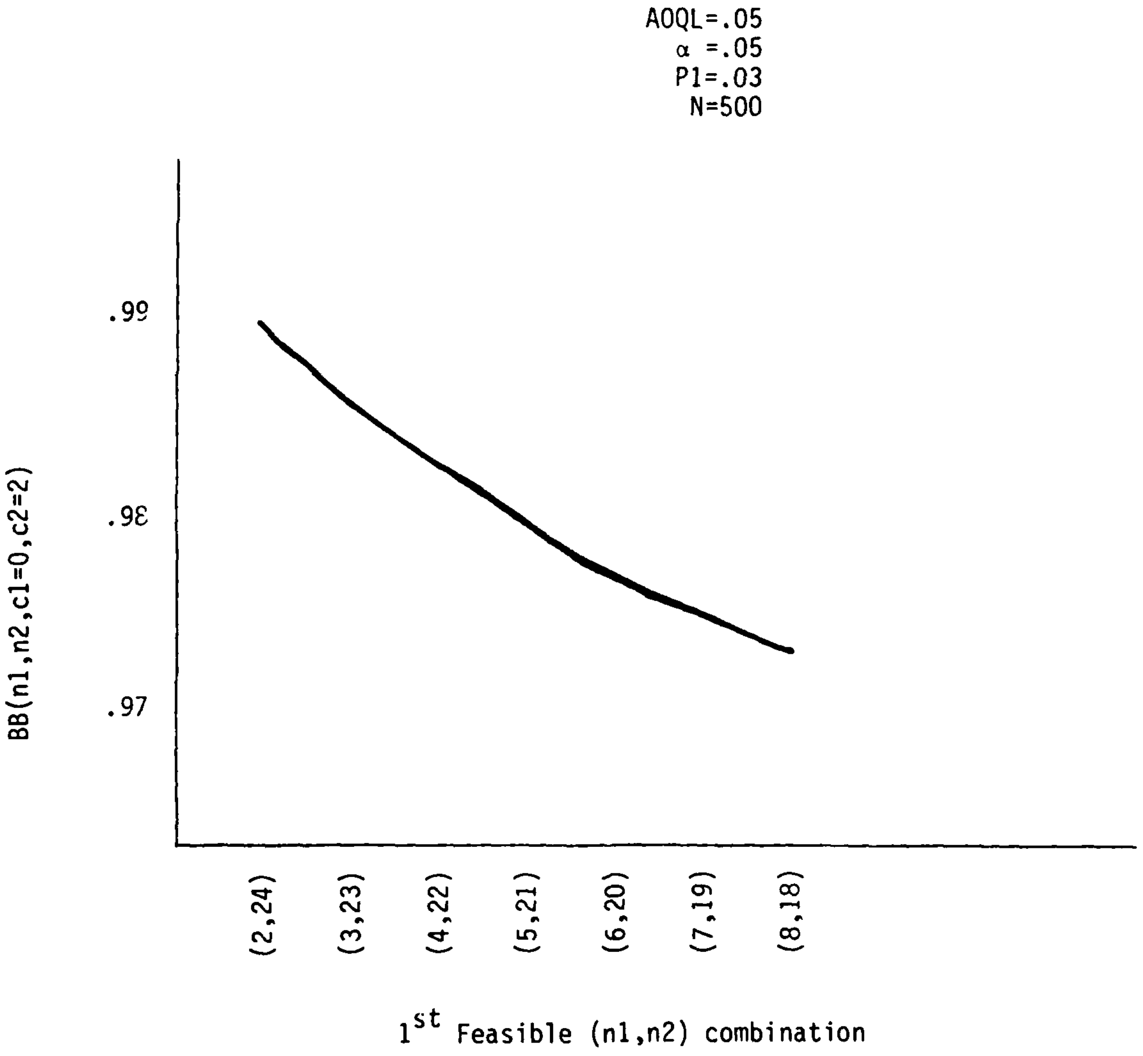


FIGURE 8. Values of L(p) for First Feasible n1,n2 Combinations

minimum), is larger than the preceeding one. Thus, the minimum sampling plan for the design parameters is found.

The computer code was written in Fortran IV and run on the PDP 11/34 digital computer. For further clarification of the computer code, flowcharts of the main program and the subroutines that compute p* and AOQL are included in the appendix as well as a listing of the code.

COMPARATIVE RESULTS

The computer program was run with various AOQL values. The following table characterizes the effectiveness of the program. Sampling plans derived from a study that minimizes the ASN(p1), plans derived by Dodge and Romig, and MIL-STD-105D plans are shown for comparison.

The values of n1 and n2 given in the Dodge Romig tables [1] are based on the largest lot size in each range. The values of c1 and c2 correspond to the mean lot size and to the mean value of the process average. This ensures that the AOQL values over the range will not exceed the specified value, however this only gives the average most economical plan within a range of process averages and lot sizes. The computed sampling plan will vary as the lot size and design process average change. For example, for AOQL=1%, α=0.03465, p1=0.0065, and a lot size of 301 (lower bound of the Dodge-Romig range), the computer program gives a sampling plan of c1=0, c2=2, n1=40, and n2=84 with an AFI(p1)=0.2170. At N=400, the upper bound of the lot size range, c1=0, c2=2, n1=46, n2=73, an AFI(p1)=0.1863. Dodge and Romig give the optimal sampling plan as c1=0, c2=2, n1=55, n2=60.

TABLE III

Comparisons of Computed Results with Other Standard Plans.

Parameters	Plans using minimum AFI criterion	Plans using Minimum ASN criterion	Dodge-Romig plans (Min AFI)	MIL-STD-105D Normal-II	MIL-STD-105D Tightened-II
AOQL=.01 α =.0347 β =.0736 p1=.0065 p2=.035 N=350	C1=0 c2=2 n1=43 n2=79 AFI=.2005	C1=0 c2=2 n1=56 n2=58	C1=0 c2=2 n1=55 n2=60 N=301-400 AFI=.2321	c1=0 c2=1 r1=2 r2=2 n1=50 n2=50 AFI=.2596	c1=0 c2=1 r1=2 r2=2 n1=80 n2=80 AFI=.4421
AOQL=.02 α =.0381 β =.2029 p1=.015 P2=.067 N=350	c1=0 c2=3 n1=26 n2=55 AFI=.1465	c1=0 c2=3 n1=35 n2=52	c1=0 c2=3 n1=33 n2=55 N=301-400 AFI=.184522	c1=0 c2=3 r1=3 r2=4 n1=32 n2=32 AFI=.1387	c1=0 c2=1 r1=2 r2=2 n1=32 n2=32 AFI=.2885
AOQL=.025 α =.0389 β =.0844 p1=.025 p2=.09 N=500	c1=1 c2=6 n1=42 n2=76 AFI=.145115	C1=1 c2=6 n1=50 n2=80	C1=1 c2=6 n1=50 n2=80 N=401-500 AFI=.1858	c1=1 c2=4 r1=4 r2=5 n1=32 n2=32 AFI=.0924	c1=0 c2=3 r1=3 r2=4 n1=32 n2=32 AFI=.1630
AOQL=.05 α =.0433 β =.1062 p1=.04 p2=.165 N=450	c1=0 c2=5 N1=11 n2=46 AFI=.078748	C1=1 c2=4 N1=25 n2=25	C1=1 c2=4 N1=16 n2=34 N=401-500 AFI=.0700	c1=2 c2=6 r1=5 r2=7 n1=32 n2=32 AFI=.0904	c1=1 c2=4 r1=4 R2=5 n1=32 n2=32 AFI=.1805
AOQL=.05 α =.0442 β =.0650 p1=.04 p2=.18 N=1500	c1=1 c2=11 n1=22 n2=111 AFI=.03524	c1=1 c2=4 n1=26 n2=24	c1=0 c2=4 n1=17 n2=33 N=1001-2000 AFI=.0651	c1=5 c2=12 r1=9 r2=13 n1=80 n2=80 AFI=.0673	c1=3 c2=11 r1=7 r2=12 n1=80 n2=80 AFI=.0982
AOQL=.05 α =.0405 β =.0987 P1=.04 p2=.17 N=350	c1=0 c2=5 n1=11 n2=44 AFI=.091434	c1=1 c2=4 n1=25 n2=24	c1=0 c2=4 n1=16 n2=33 N=301-400 AFI=.125809	c1=2 c2=6 r1=5 r2=7 n1=32 n2=32 AFI=.113002	c1=1 c2=4 r1=4 r2=5 n1=32 n2=32 AFI=.2043

The computer results for a 2% AOQL with α=0.0478, p1=0.02 and N=1500 follow:

c1	c2	n1	n2	AFI(p1)
0	7	28	128	.095858
1	7	54	144	.093983
2	7	81	116	.101264
0	8	28	193	.096234
1	8	55	166	.092043
2	8	81	142	.098126
0	9	28	217	.097668
1	9	55	190	.091320
2	9	82	163	.096105
0	10	28	242	.101074
1	10	55	214	.092049
2	10	82	187	.095498

Dodge and Romig give the optimal plan of c1=1, c2=8, n1=80, n2=160 with the AFI(p1) being 0.1444. The minimum sampling plan found above (c1*=1, c2*=9, n1*=55, n2*=190) has a lower AFI(p1) value than that corresponding to c1=1, c2=8. The difference is small; however this may account for some of the discrepancies in the results.

The sampling plans derived from large lot sizes do not correspond well with the Dodge-Romig plans. For the parameters AOQL=0.05, α=0.0442, p1=0.04 and N=1500,

AFI based algorithm plans	Dodge-Romig AOQL plans	MIL-STD-105D Normal-II	MIL-STD-105D Tightened-II
c1=1	c1=0	c1=5	c1=3
c2=11	c2=4	c2=12	c2=11
n1=22	n1=17	r1=9	R1=7
n2=111	n2=33	r2=13	R2=12
		n1=80	n1=80
		b2=80	n2=80

The c* value of the single sampling plan based on the AFI criterion is 5. Since c2 is initially set to c*, the acceptance numbers found by the AFI algorithm and those of the Dodge-Romig plans would never be the same. It appears that for large values of N, the minimum AFI plans resulting from the described program more closely resemble those of the MIL-STD-105D plans.

The results using the minimum AFI(p1) criterion are very similar to those using a minimum ASN criterion. The AFI based plans and the Dodge-Romig plans deal with rectifying inspection whereas the minimum ASN based plans do not.

In Table III, note that the total number sampled, n1+n2, is very close in the AFI algorithm, ASN algorithm, and Dodge-Romig plans. The individual values of n1 and n2 vary to a larger degree than the total. The n1 value from the ASN algorithm plan is greater than that of the Dodge-Romig plans and n2 is less than the n2 value of the Dodge-Romig plans. Also, n1(AFI) is less than n1(Dodge-Romig) and n2(AFI) is greater than n2(Dodge-Romig).

The MIL-STD-105D tables specify a fixed sample for a given lot size. In most cases, the total number sampled does not correspond well to the other plans. MIL-STD Level II plans for Normal and Tightened inspection are included in the table. It is not clear which specifications most closely related to the other sampling plans. In some instances the Normal inspection plans resemble the other plans, and in other cases the Tightened inspection plans are more similar.

The computer based technique is advantangeous in that plans corresponding to specific AOQL values, lot sizes, and process averages can be found directly rather than relying on ranges of lot size and process averages which can yield only approximate optimal plans.

REFERENCES

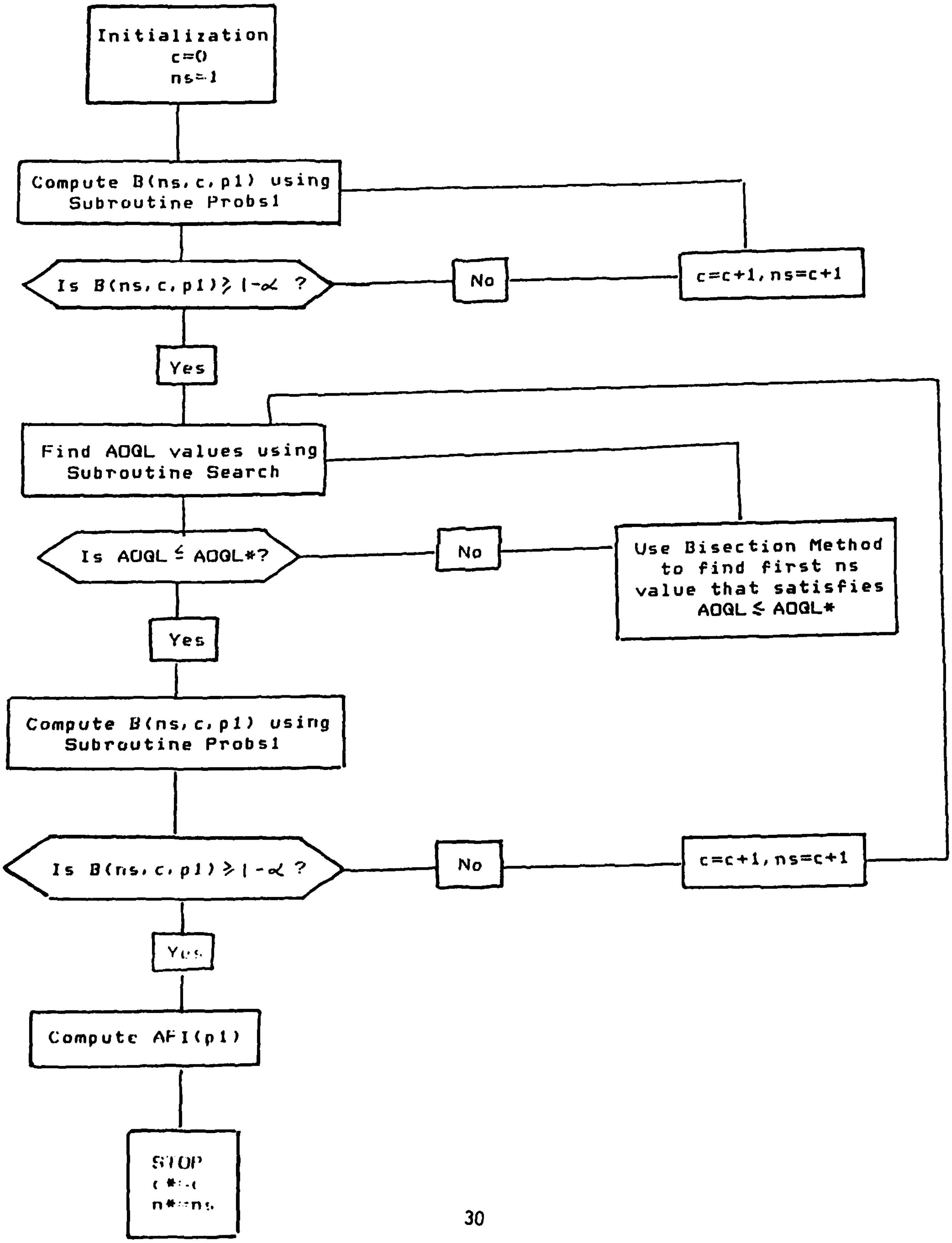
1. Dodge, H. F. and Romig, H. G., Sampling Inspection Tables, John Wiley & Sons, Inc., 1959.

2. Grant, E. L. and Leavenworth, R. S. Statistical Quality Control, McGraw-Hill Book Co., 5th Edition, 1980.

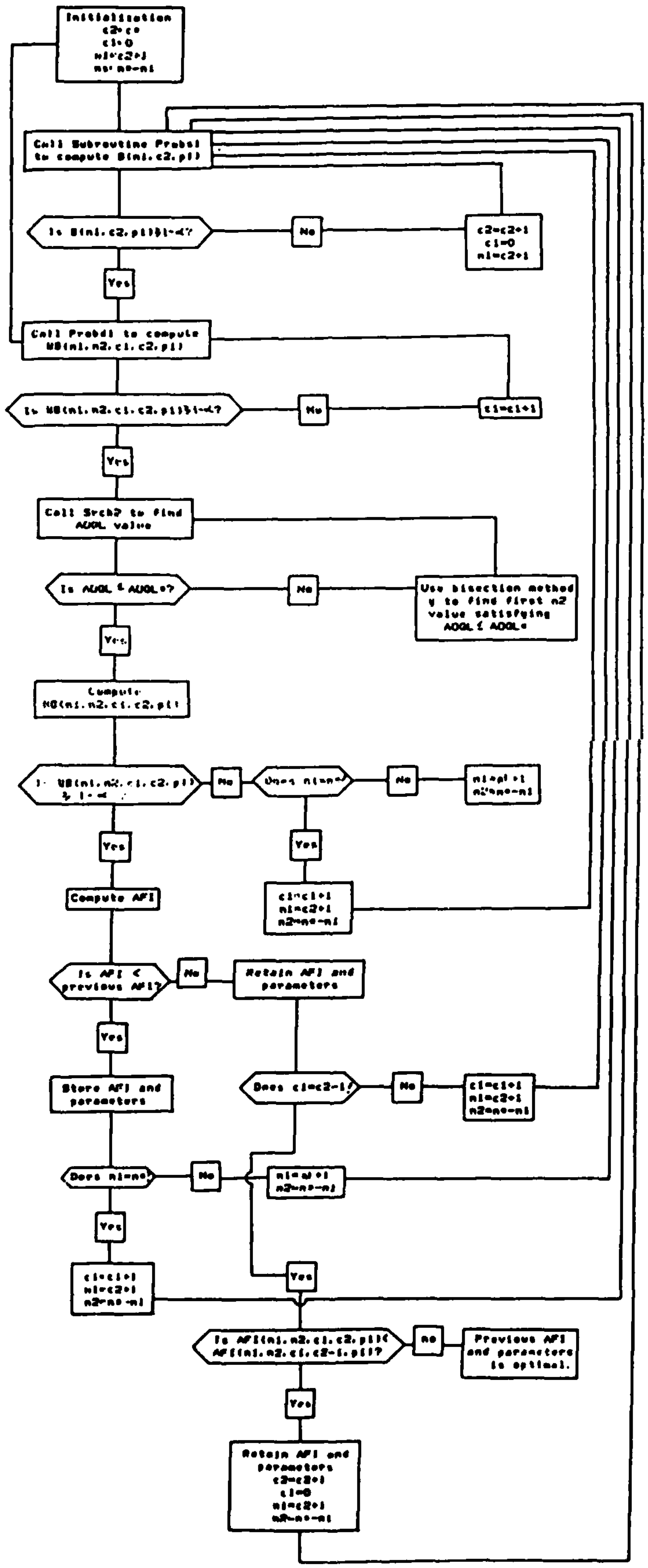
3. Wagner, H. W., Principles of Operations Research, Prentice-Hall, Inc. 1969.

APPENDIX

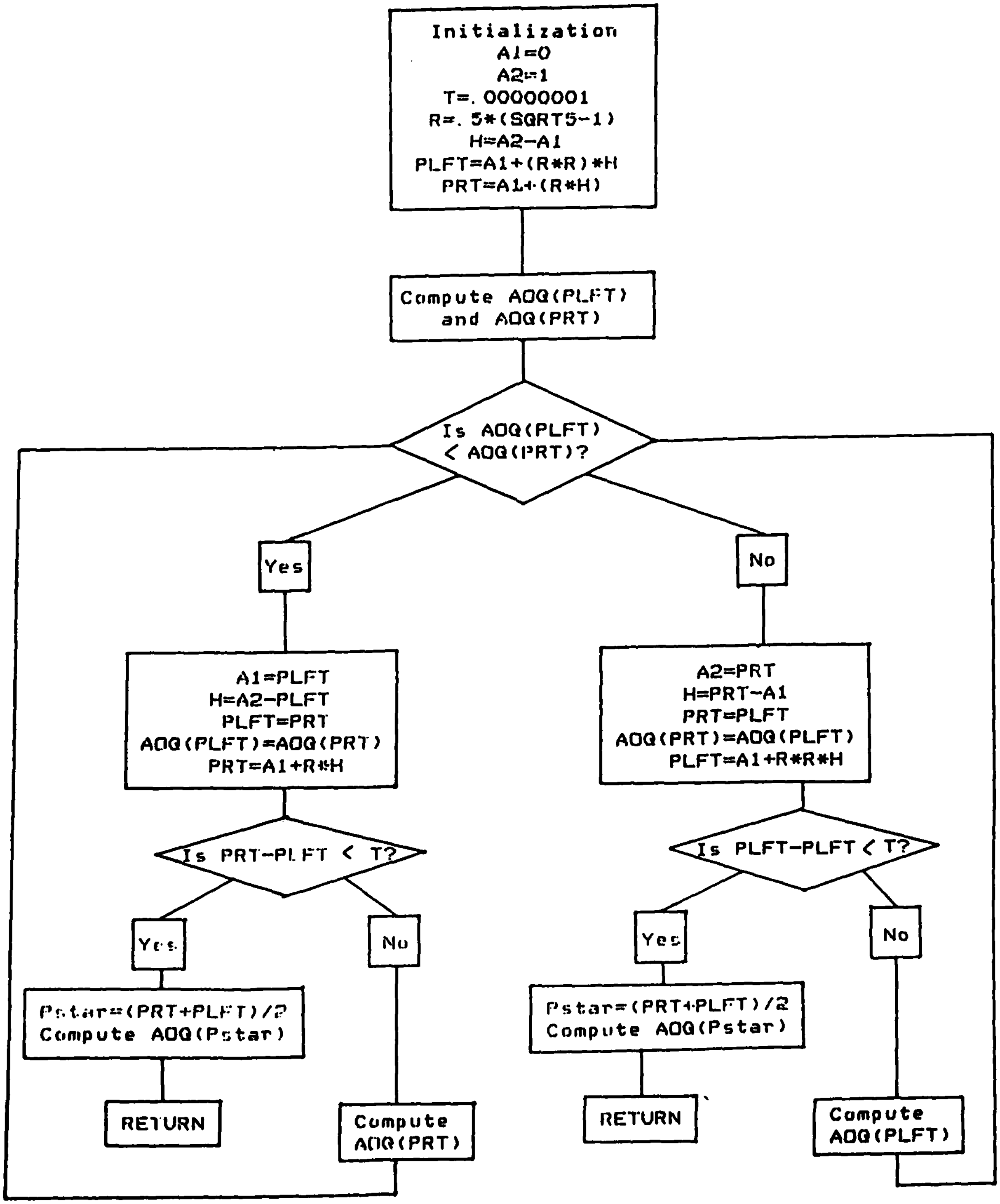
SINGLE SAMPLING FLOW CHART



DOUBLE SAMPLING FLOW CHARTS



SUBROUTINE SEARCH



QAFI.FOR

```
C
C**************************************************
C       QUALITY CONTROL PROGRAM TO DERIVE DOUBLE SAMPLING
C       PLANS TO MINIMIZE AVERAGE FRACTION INSPECTED.
C
C       PROGRAMMED BY JO ELLEN WALKER
C       DEPARTMENT OF INDUSTRIAL AND SYSTEMS ENGINEERING
C       UNIVERSITY OF FLORIDA
C       GAINESVILLE, FLORIDA  32611
C**************************************************
C
        SUBROUTINE QAFI
        INTEGER C1,C2,C,CSTAR,C2M1
        INTEGER R1,R1M1
        DOUBLE PRECISION SUMLOG
        BYTE OUTFIL(8)
        COMMON/BLK4/ALPHA,BETA
        COMMON/BLK6/C1,C2
        COMMON/BLK7/SUMLOG(4000)
        COMMON/BLK8/N
        COMMON/BLK12/OUTFIL
C**************************************************
C       INPUT PARAMETERS
C**************************************************
        KINDEX=1
        WRITE(5,10)
     10 FORMAT(10X,'ENTER VALUE OF ALPHA')
        READ(5,*)ALPHA
        WRITE(5,15)
     15 FORMAT(10X,'ENTER VALUE OF PO')
        READ(5,*)PO
        WRITE(5,20)
     20 FORMAT(10X,'ENTER AOQL VALUE')
        READ(5,*)AOQL
        WRITE(5,25)
     25 FORMAT(10X,'ENTER LOT SIZE')
        READ(5,*)NNN
C
C
C
        WRITE(1,28)
     28 FORMAT('1',///10X,'DEPT. OF ISE  '
       $/,10X,'UNIVERSITY OF FLORIDA '/
       $/5X,5('*'),'DOUBLE SAMPLING SYSTEM TO MINIMIZE AFI',
       $5('*'),2X,/)
        WRITE(5,30)ALPHA
        WRITE(1,30)ALPHA
     30 FORMAT(10X,'ALPHA=',2X,F8.6)
        WRITE(5,35)PO
        WRITE(1,35)PO
     35 FORMAT(10X,'PO=',2X,F8.6)
        WRITE(5,40)AOQL
        WRITE(1,40)AOQL
     40 FORMAT(10X,'AOQL=',2X,F8.6)
        WRITE(5,45)NNN
        WRITE(1,45)NNN
     45 FORMAT(10X,'N=',2X,I6)
```

QAFI

```
C****************************************************
C       COMPUTE SINGLE SAMPLING PLAN
C****************************************************
C
C       INITIALIZATION
C
C****************************************************
        NS=1
        C=0
        N=0
        AFI=1.DO
C
        WRITE(5,50)
        WRITE(1,50)
     50 FORMAT(///10X,'*** SINGLE SAMPLING PLAN ***')
C****************************************************
C       FIND NS,C COMBO THAT SATISFIES L(PO) G.T. 1-ALPHA
C****************************************************
     55 CALL PROBS1(NS,PO,C,BXLEC)
        IF(BXLEC.LT.(1.DO-ALPHA))C=C+1
        IF(BXLEC.LT.(1.DO-ALPHA))NS=C+1
C****************************************************
C       SEARCH TO FIND MIN NS VALUE SUCH THAT AOQL L.T. AOQL*
C****************************************************
     60 CALL SEARCH(NNN,C,NS,SAOQ)
        IF(SAOQ.LE.AOQL)GOTO 75
        NSTEMP=NNN
        CALL SEARCH(NNN,C,NSTEMP,SAOQ)
        IF(SAOQ.GT.AOQL)C=C+1
C
C
        IF(SAOQ.GT.AOQL)NS=C+1
C
C
        BL=NS
        BH=NNN
     65 NSTEMP=IIDINT((BL+BH)/2.DO)
        CALL SEARCH(NNN,C,NSTEMP,SAOQ)
        IF(SAOQ.LE.AOQL)BH=NSTEMP
C
        IF(SAOQ.LE.AOQL)AOQ=SAOQ
C
        IF(SAOQ.GT.AOQL)BL=NSTEMP
        IF((BH-BL).EQ.1.DO)GOTO 70
        GOTO 65
     70 NS=NSTEMP
        IF(SAOQ.GT.AOQL)NS=BH
C****************************************************
C       CHECK THAT NS,C COMBO STILL SATISFIES L(PO) CONSTRAINT
C****************************************************
     75 CALL PROBS1(NS,PO,C,BXLEC)
        IF(BXLEC.GE.(1.DO-ALPHA))GOTO 80
        C=C+1
C
        NS=C+1
C
        GOTO 60
```

```
C*****************************************************
C       COMPUTE AFI
C*****************************************************
   80 ATIPO=NS*BXLEC+NNN*(1.DO-BXLEC)
      AFIPO=ATIPO/NNN
C
      NSTAR=NS
C
      WRITE(5,85)NS
      WRITE(1,85)NS
   85 FORMAT(//10X,'NS=',I3)
      WRITE(5,90)C
      WRITE(1,90)C
   90 FORMAT(10X,'C=',I2)
      WRITE(5,95)AFIPO
      WRITE(1,95)AFIPO
   95 FORMAT(10X,'AFI(PO)=',F8.6)
C
  100 CSTAR=C
C*****************************************************
C       DOUBLE SAMPLING
C*****************************************************
C
      WRITE(5,105)
      WRITE(1,105)
  105 FORMAT(///10X,'** DOUBLE SAMPLING PLANS **')
      PLAN=1.DO
      C2=CSTAR
  110 R1=C2+1
      C2M1=C2-1
      C1PO=C1+1
      R1M1=R1-1
      DDATI=NNN
      ITCMIN=1.
C
      C1=0
      JJ=0
C
      WRITE(5,115)
      WRITE(1,115)
  115 FORMAT(///10X,'C1',6X,'C2',7X,'N1',8X,'N2',9X,'AFI',//)
C*****************************************************
C       CALCULATE FIRST SAMPLE NUMBER
C
C       FROM RESULTS OF PREVIOUS RUNS, IT WAS FOUND THAT N1 IS NOT LESS
C       THAN NSTAR/8.  THUS, THE INITIAL VALUE OF N1 IS SET ACCORDINGLY.
C*****************************************************
C
  120 DO 165 LL=INT(NSTAR/8),NSTAR
      N1=LL
      IF(N1.LT.C2)N1=R1M1
C*****************************************************
C       CHECK B(N1,PO,C2) G.T. 1-ALPHA CONSTRAINT
C*****************************************************
  125 CALL PROBS1(N1,PO,C2,BXLEC)
      IF(BXLEC.LT.(1.-ALPHA))GOTO 175
C*****************************************************
```

```
C      CALCULATE SECOND SAMPLE
C*****************************  ************************
       N2=NSTAR-N1
C******************************************************
C      CHECK THAT DOUBLE PROBABILITY G.T. 1-ALPHA
C******************************************************
   130 CALL PROBD1(N1,N2,PO,DPROB,KINDEX,R1)
       IF(DPROB.GE.(1.-ALPHA))GOTO 135
       IF(C1.EQ.C2M1)GOTO 175
       C1=C1+1
       GOTO 130
C******************************************************
C      FIND N2 VALUE SATISFYING AOQL L.T. AOQL*
C******************************************************
   135 CALL SRCH2(NNN,N1,N2,AOQ1)
       IF(AOQ1.LE.AOQL)GOTO 150
C******************************************************
C      N2 WILL NOT BE LESS THAN N1*9, THE INITIAL LOWER BOUND ON N2.
C******************************************************
       N2TEMP=N1*9
C
       CALL SRCH2(NNN,N1,N2TEMP,AOQ1)
       IF(AOQ1.GT.AOQL)GOTO 165
C
       BL=N2
       BH=N1*9
   140 N2TEMP=INT((BL+BH)/2)
       CALL SRCH2(NNN,N1,N2TEMP,AOQ1)
       IF(AOQ1.LE.AOQL)BH=N2TEMP
       IF(AOQ1.LE.AOQL)FAOQ=AOQ1
C
       IF(AOQ1.GT.AOQL)BL=N2TEMP
       IF((BH-BL).EQ.1.)GOTO 145
       GOTO 140
   145 N2=N2TEMP
       IF(AOQ1.GT.AOQL)N2=BH
C******************************************************
C      CHECK THAT BINOMIAL PROBABILITIES ARE G.T. 1-ALPHA
C******************************************************
       CALL PROBD1(N1,N2,PO,DPROB,KINDEX,R1)
       IF(DPROB.LT.(1.-ALPHA))GOTO 165
   150 CALL PROBS1(N1,PO,C1,BXLEC)
       PA2=DPROB-BXLEC
C******************************************************
C      COMPUTE ATI
C******************************************************
       DATI=N1*DPROB+N2*PA2+NNN*(1.-DPROB)
C******************************************************
C      IF THE ATI INCREASES, CONTINUE FOR 5 ADDITIONAL INCREASING
C      ITERATIONS.  THEN INCREMENT C1 AND CONTINUE.
C******************************************************
       IF(JJ.EQ.4)GOTO 155
       IF(DATI.GE.DDATI)JJ=JJ+1
       IF(DATI.GE.DDATI)GOTO 165
       DDATI=DATI
       DDAOQ=FAOQ
C
```

```
      K1=C1
      K2=C2
      K3=N1
      K4=N2
C
      GOTO 165
C*****************************************************
C     MINIMUM OF CELL (TCMIN) FOUND
C*****************************************************
  155 TCMIN=DDATI/NNN
      DDATI=NNN
C
      WRITE(5,160)K1,K2,K3,K4,TCMIN
      WRITE(1,160)K1,K2,K3,K4,TCMIN
  160 FORMAT(10X,I2,6X,I2,5X,I4,6X,I4,7X,F8.6)
C*****************************************************
C     IF MINIMUM OF COLUMN IS FOUND, INCREASE C2
C*****************************************************
      IF(TCMIN.GE.TTCMIN)GOTO 170
      KK1=K1
      KK2=K2
      KK3=K3
      KK4=K4
      TTCMIN=TCMIN
      TTAOQ=DDAOQ
C
      IF (C1.EQ.C2M1) GOTO 175
      C1=C1+1
C
      JJ=0
      N1=KK3
C
      GOTO 125
C
  165 CONTINUE
C*****************************************************
C     MINIMUM OF COLUMN (TMIN) FOUND
C*****************************************************
  170 TMIN=TTCMIN
      TAOQ=TTAOQ
C*****************************************************
C     IF MINIMUM SAMPLING PLAN FOUND, STOP
C*****************************************************
      IF(TMIN.GE.PLAN)GOTO 190
C
      PLAN=TMIN
      KKK1=KK1
      KKK2=KK2
      KKK3=KK3
      KKK4=KK4
  175 C2=C2+1
C*****************************************************
C      NEW BOUNDS ON SAMPLING PLAN CALCULATED FOR NEW VALUE OF C2
C*****************************************************
  180 CALL SEARCH(NNN,C2,NSTAR,AOQ)
      IF(AOQ.LE.AOQL)GOTO 185
      IF(NSTAR.GT.NNN)GOTO 175
```

```
      NSTAR=NSTAR+1
      GOTO 180
C
C
  185 CALL PROBS1(NSTAR,PO,C2,BXLEC)
      GOTO 110
C
C
  190 WRITE(5,195)
      WRITE(1,195)
  195 FORMAT(//10X,'SAMPLING PLAN MINIMUMS')
      WRITE(5,200)KKK1,KKK2
      WRITE(1,200)KKK1,KKK2
  200 FORMAT(/10X,'C1=',I2,2X,'C2=',I2)
      WRITE(5,205)KKK3,KKK4
      WRITE(1,205)KKK3,KKK4
  205 FORMAT(10X,'N1=',I3,2X,'N2=',I3)
      WRITE(5,210)PLAN
      WRITE(1,210)PLAN
  210 FORMAT(10X,'MINIMUM AFI=',F8.6)
  215 RETURN
      END
```

```
      SUBROUTINE SEARCH(NNN,C,NS,AOQ)
C**************************************************************
C     SEARCH TO FIND VALUE OF PSTAR USING GOLDEN
C     SECTION METHOD. INITIAL LIMITS OF  0 AND 1
C**************************************************************
      INTEGER C
      DOUBLE PRECISION SUMLOG
      COMMON/BLK7/SUMLOG(4000)
      COMMON/BLK8/N
C
      A1=0.D0
      A2=1.D0
      T=1.D-3
      R=5.D-1*(DSQRT(5.D0)-1.D0)
      H=A2-A1
      PLFT=A1+(R*R)*H
      PRT=A1+(R*H)
C
C
      CALL PROBS1(NS,PLFT,C,BXLEC)
      ATI=(NS*BXLEC)+NNN*(1.D0-BXLEC)
      AFI1=ATI/NNN
      AOQ1=PLFT*(1.D0-AFI1)
      CALL PROBS1(NS,PRT,C,BXLEC)
      ATI=(NS*BXLEC)+NNN*(1.D0-BXLEC)
      AFI2=ATI/NNN
      AOQ2=PRT*(1.D0-AFI2)
      GOTO 110
C
C
  100 CALL PROBS1(NS,PLFT,C,BXLEC)
      ATI=(NS*BXLEC)+NNN*(1.D0-BXLEC)
      AFI1=ATI/NNN
      AOQ1=PLFT*(1.D0-AFI1)
      GO TO 110
C
  105 CALL PROBS1(NS,PRT,C,BXLEC)
      ATI=(NS*BXLEC)+NNN*(1.D0-BXLEC)
      AFI2=ATI/NNN
      AOQ2=PRT*(1.D0-AFI2)
C
  110 IF(AOQ1.LT.AOQ2) GOTO 115
      A2=PRT
      H=PRT-A1
      IF(ABS(PRT-PLFT).LE.T)GOTO 120
C
      PRT=PLFT
      PLFT=A1+(R*R)*H
      AOQ2=AOQ1
      GO TO 100
  115 A1=PLFT
      H=A2-PLFT
C
      IF(ABS(PRT-PLFT).LE.T)GOTO 120
      PLFT=PRT
      PRT=A1+R*H
      AOQ1=AOQ2
```

```
          GO TO 105
C
C
      120 PS=(PLFT+PRT)/2.D0
          CALL PROBS1(NS,PS,C,BXLEC)
          AFI=((NS*BXLEC)+NNN*(1.D0-BXLEC))/NNN
          AOQ=PS*(1.D0-AFI)
          RETURN
          END
```

```
      SUBROUTINE SRCH2(NNN,N1,N2,AOQ)
C
      INTEGER C1,C2,R1
      COMMON/BLK6/C1,C2
C
      KINDEX=1
      A1=0.
      A2=1.
      T=.0001
      R=.5*(DSQRT(5.D0)-1.)
      H=A2-A1
      PLFT=A1+(R*R)*H
      PRT=A1+(R*H)
C
C
      CALL PROBS1(N1,PLFT,C1,PA1)
      CALL PROBD1(N1,N2,PLFT,DPROB,KINDEX,R1)
      PA2=DPROB-PA1
      ATI=DPROB*N1+PA2*N2+NNN*(1.-DPROB)
      AFI1=ATI/NNN
      AOQ1=PLFT*(1.-AFI1)
      CALL PROBS1(N1,PRT,C1,PA1)
      CALL PROBD1(N1,N2,PRT,DPROB,KINDEX,R1)
      PA2=DPROB-PA1
      ATI=DPROB*N1+PA2*N2+NNN*(1.-DPROB)
      AFI2=ATI/NNN
      AOQ2=PRT*(1.-AFI2)
      GOTO 110
C
C
  100 CALL PROBS1(N1,PLFT,C1,PA1)
      CALL PROBD1(N1,N2,PLFT,DPROB,KINDEX,R1)
      PA2=DPROB-PA1
      ATI=DPROB*N1+PA2*N2+NNN*(1.-DPROB)
      AFI1=ATI/NNN
      AOQ1=PLFT*(1.-AFI1)
      GOTO 110
  105 CALL PROBS1(N1,PRT,C1,PA1)
      CALL PROBD1(N1,N2,PRT,DPROB,KINDEX,R1)
      PA2=DPROB-PA1
      ATI=DPROB*N1+PA2*N2+NNN*(1.-DPROB)
      AFI2=ATI/NNN
      AOQ2=PRT*(1.-AFI2)
C
C
  110 IF(AOQ1.LT.AOQ2)GOTO 115
C
C
      A2=PRT
      H=PRT-A1
      IF(ABS(PRT-PLFT).LE.T)GOTO 120
      PRT=PLFT
      PLFT=A1+(R*R)*H
      AOQ2=AOQ1
      GOTO 100
  115 A1=PLFT
C
```

```
C
      PA2=PLFT
      IF(ABS(PRT-PLFT).LE.T)GOTO 120
      PLFT=PRT
      PRT=A1+R*H
      AOQ1=AOQ2
      GOTO 105
C
C
  120 PS=(PLFT+PRT)/2.
      CALL PROBS1(N1,PS,C1,PA1)
      CALL PROBD1(N1,N2,PS,DPROB,KINDEX,R1)
      PA2=DPROB-PA1
      AFI=(DPROB*N1+PA2*N2+NNN*(1.-DPROB))/NNN
      AOQ=PS*(1.-AFI)
      RETURN
      END
```

```
      SUBROUTINE PROBS1(NN,P,C,BXLEC)
C**************************************************
C     THIS SUBROUTINE COMPUTES CUMULATIVE BINOMIAL
C     PROBABILITIES
C**************************************************
      INTEGER C
      DOUBLE PRECISION SUMLOG
C
      COMMON/BLK7/SUMLOG(4000)
      COMMON/BLK8/N
C
      Q=1.-P
C**************************************************
C           BINOMIAL PROB. WHEN C=0
C**************************************************
      CSUMS=Q**NN
      IF (C.EQ.0) GOTO 45
C**************************************************
C      AVOID RECOMPUTING SUMLOG(I)'S ALREADY IN MEMORY
C**************************************************
      IF (N-NN) 10,25,25
   10 M=N+1
C**************************************************
C            COMPUTE N SUMLOGS-EQUIVALENT TO N-FACTORIAL
C**************************************************
      IF (M.GT.1) GOTO 15
      SUMLOG(1)=0.
      IF(NN.LE.1) GOTO 25
      M=2
   15 DO 20 I=M,NN
          SUMLOG(I)=DLOG10(DFLOAT(I))+SUMLOG(I-1)
   20 CONTINUE
C**************************************************
C            COMPUTE C SUMS-EQUIVALENT TO SSUM OF PROB.COMPIN.
C            I.E. CUMULATIVE BINOMIAL DISTRIBUTION COMPUTATION
C**************************************************
   25 IF(NN.GT.N) N=NN
C**************************************************
C            DETERMINE BEST NUMBER HANDLING LOOP
C**************************************************
      IF (NN.GT.300) GOTO 35
      DO 30 K=1,C
          CSUMS=10.**(SUMLOG(NN)-SUMLOG(NN-K)-SUMLOG(K))
     1                      *P**K*Q**(NN-K)+CSUMS
   30 CONTINUE
      GOTO 45
C**************************************************
C            LOOP FOR LARGE EXPONENTS
C**************************************************
   35 DO 40 K=1,C
          CSUMS=10.**(SUMLOG(NN)-SUMLOG(NN-K)-SUMLOG(K)
     1      +K*DLOG10(DBLE(P))+(NN-K)*DLOG10(DBLE(Q)))+CSUMS
   40 CONTINUE
C
   45 BXLEC = CSUMS
      RETURN
      END
```

```
      SUBROUTINE PROBD1(N1,N2,P,DPROB,K,R1)
C**************************************************
C     THIS SUBROUTINE COMPUTES DOUBLE PROBABILITIES FOR
C     COMPUTING SECOND SAMPLE NUMBER OF DOUBLE SAMPLING NUMBER
C**************************************************
      COMMON/BLK6/C1,C2
      INTEGER C1,C2,R1
C
      IF(K.EQ.1) CALL PROBS1(N1,P,C1,BXLEC)
      IF(K.EQ.2) CALL PROBS2(N1,P,C1,BXLEC)
      DPROB=BXLEC
      TEMP=BXLEC
      NTEMP=C1+1
      KTEMP=R1-1
      DO 10 IX=NTEMP,KTEMP
         I=IX
         J=C2-I
         IF(K.EQ.1) CALL PROBS1(N1,P,I,BXLEC)
         IF(K.EQ.2) CALL PROBS2(N1,P,I,BXLEC)
         PROB1=BXLEC-TEMP
         TEMP=BXLEC
         IF(K.EQ.1) CALL PROBS1(N2,P,J,BXLEC)
         IF(K.EQ.2) CALL PROBS2(N2,P,J,BXLEC)
         DPROB=DPROB+(PROB1*BXLEC)
   10 CONTINUE
C
      RETURN
      END
```

ATE
LMED
8